WHAT IS CLAIMED IS:

1. A method for automated inference and construction of Extensible Markup Language (XML) structure from textual documents, the method comprising:

identifying a target document type from a set of textual documents with generally consistent inherent logical structure and formatting;

defining an XML schema that models the logical structure inherent in the target document type;

creating a structure inference definition for the target document type, comprising a multiplicity of baseline element definitions, baseline elements being select leaf-level or near-leaf-level elements from the target document type encompassing the bulk of the document text;

defining recognition patterns for the baseline elements; and

invoking a computer-executable engine component to apply the structure inference definition to one or more instances of the target document type to produce XML structure relating to the defined schema, the operation of said component comprising: parsing a given document; recognizing instances of the designated baseline elements via pattern search and matching; and inferring and constructing higher-level element structure in best possible agreement with the declared content models of all ancestor elements .

2. A method as recited in claim 1 wherein a baseline element is identified by a schema path, comprising a sequence of one or more XML element or element type steps, the first step designating a global schema element or type and each subsequent step designating a child element or element group of its predecessor.

3. A method as recited in claim 1 wherein the baseline element recognition patterns comprise: text patterns such as literals, wildcards, and regular expressions; formatting patterns such as font style, font name, font size, composite style name, paragraph indentation or outline level; and logical compositions of atomic text and formatting patterns and pattern groups.

4. A method as recited in claim 1 further comprising defining additional patterns and structure inference and construction rules for one or more levels of nested elements in the context of a designated baseline element, said patterns and rules being used by the structure inference and construction engine to produce nested element structure within the text range and schema context of a matched baseline element.

5. A method as recited in claim 1 wherein a baseline element pattern comprises the following pattern components, each of which may be selectively included or omitted by the structure inference definition designer, and when omitted, appropriate matching semantics are ascribed to the baseline element pattern as a whole:

a leading pattern, intended to match the document range immediately preceding the content range of the baseline element, allowing intervening whitespace;

a content pattern, intended to match what would become the content range of the baseline element; and

a trailing pattern, intended to match the document range immediately following the matched content range for a baseline element, allowing intervening whitespace, and end document position of such match serving as starting position for matching the patterns of the following baseline elements.

6. A method as recited in claim 1 wherein the defining of recognition patterns for the baseline elements comprises assigning a priority value or pattern weight value which influences the selection of one baseline element when the patterns for more than one element yield competing/ambiguous matches at a particular document position.

7. A method as recited in claim 1 wherein the structure inference and construction engine compiles and uses a baseline element finite state machine (BESM) to minimize ambiguity and false matches by considering only a small number of expected baseline elements at a given document position, said state machine being derived by recursive aggregation of all schema element content models, starting from the designated root element, down to the level of designated baseline elements, and incorporating in its transition labels the identifies and specific instance contexts of baseline elements.

8. A method as recited in claim 1 wherein the invoking of the structure inference and construction engine is triggered automatically when a new document file is detected in a predefined incoming document folder or when a document is received via an Application Programming Interface (API) from an external client component, and the resultant XML document is saved in a predefined output folder or returned to the API client, respectively.

9. A method as recited in claim 1 wherein the structure inference and construction component is configured to operate in unattended batch mode, on multiple documents sequentially or in parallel.

10. A method for applying XML-compatible markup to unstructured textual documents, comprising:

defining an XML schema in accordance with which documents are to be marked up;
opening a given target document in a host Application Programming Interface (API) enabled generic wordprocessor application capable of storing XML-compatible non-native markup in its documents;
using the API of the host application to parse the document content and to perform element pattern matching and yielding XML structure in accordance with the chosen schema; and
storing the inferred structure within the document as XML-compatible markup via the API of the host application.

11. A method as recited in claim 10 wherein said using step comprises a structure inference method for parsing a given document; recognizing instances of designated baseline elements via pattern search and matching; and inferring and constructing higher-level element structure in best possible conformance with the defined XML schema.

12. A method as recited in claim 10 wherein the original visual formatting and textual content of the target document remain intact after applying XML markup to it.

13. A method as recited in claim 10 wherein the XML structure inference and markup creation are limited to a select range or number of select ranges of the target document.

14. A method as recited in claim 10 wherein a structure inference definition for the chosen XML schema is created by means of dedicated Graphical User Interface (GUI) integrated in the GUI workspace of the host application.

15. A method as recited in claim 10 further comprising presenting the user with a GUI means to review probable trouble spots in the document and to manually correct and complete the automatically generated XML markup, probable trouble spots comprising unmarked ranges, missing required elements from the XML schema, and inferred XML structure being invalid according to the schema.

16. A method for converting unstructured textual documents to XML comprising:
opening a given source document in a host Application Programming Interface (API) enabled generic wordprocessor application capable of storing XML-compatible non-native markup in its documents;
invoking a preconfigured executable component to parse the source document, map ranges of document content to XML element names defined by a custom XML schema, and apply corresponding XML-compatible markup via the API of the host application; and
using a function of the host application, via its Graphical User Interface (GUI) or API, to obtain a pure XML image of the structured document.